Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE GRAU

TÍTOL DEL TFG: Detection and Tracking over Networks

TITULACIÓ: Grau en Enginyeria de Sistemes de Telecomunicació

AUTOR: Sergi Masip

DIRECTOR: Sindri Magnússon

DATA: 19 de maig del 2016

Abstract

Knowing the location of the nodes in wireless sensor networks is essential for their operation. However, such localization in large scale networks is challenging. Additionally, the traditional localization techniques based on GPS and satellite navigation provide low accuracy and are useless indoors. Therefore, an alternative technology has to be proposed. This motivates this project's study of using wireless camera networks to track moving objects. In particular, first standard localization techniques used in a WSN are studied. Then it is explained how to detect and track objects using video cameras. Finally, an analysis is made on how video images and sensor measurements are combined in order to obtain better results. The good performance of these fusion techniques is illustrated with examples. It is demonstrated in a simulation that the localization techniques can perfectly track a moving object if there is no measurement noise. These results also demonstrate the trade-off between localization accuracy and measurement noise. Moreover, it is affirmed that increasing the number of anchor nodes, reduces the error in target localization. Finally, the detection and tracking of objects using a camera based tracking system are simulated. The general conclusion of this work is that even though video tracking systems are promising technology, for them to be a realistic option more research has to be done in signal processing, communications, computer vision and mathematical analysis.

Keywords

Wireless sensor networks, localization, wireless camera networks, detection, tracking, classification, sensor data fusion.

Acknowledgements

I would like to thank my supervisor, Sindri Magnússon, for his help and dedication during the last months.

Contents

List of Figures

List of Tables

Introduction

The first closed-circuit television (CCTV) system was installed in 1942. Since then, many improvements have been done. The increasing need of security in our society has lead to the use of more video surveillance systems. Safety and security have special importance in public places like banks, sport stadiums, supermarkets, shopping malls and parking lots. Also in public transport like airports, train stations, underground or roads and highways.

The capacity to know people's location gives a really valuable information. But in some environments it is not feasible to use satellite positioning techniques because either of its high battery consumption, low accuracy, or bad signal reception. Therefore, a substitute technology should be used. Motivated by these challenges, the aim of this project is to study the use and deployment of wireless camera networks to detect and track moving objects. Despite it is a frequently used technology, tracking moving targets using cameras is challenging. For example, the amount of information from video streams is generally huge, therefore the management and use of that information becomes important. Moreover, when there are many targets the complexity is increased because of the image processing requirements. These limitations will be discussed next.

This paper is organized as follows. In section II gives an introduction to wireless sensor networks, then explains how the localization is performed, the sensor devices used, the ranging and localization techniques and the sources of error. In section III an introduction to wireless camera networks is given, discussing how the detection, tracking and classification is done. Section IV describes the fusion of information from different sensors and gives some examples of it. In section V a new application is proposed. Section VI shows two MATLAB simulations and the results obtained. Finally in VII some conclusions are made.

Chapter 1

Localization

Network Localization is the process of estimate the position in a geographical area with the help of reference nodes. Wireless Sensor Networks (WSN) consist of a set of autonomous nodes that coordinate over wireless communication channels. Usually, those nodes collect data by sensing and monitoring the environment which they then send to neighbor nodes or to a base station. However, in most WSN tasks, the sensed data is meaningless if location information of the data source is missing. Therefore, before starting to collect information, it is essential to estimate the position of each node. Since most WSN consist of large number of nodes it is difficult for a base station to calculate the position of each one. Therefore, the nodes needs to locate themselves.

The Global Position System (GPS) is widely used for position estimation and usually provides good results outdoors. However, GPS does not give good results in indoor environments due to the signal attenuation. Moreover, using GPS for localization in WSN is generally infeasible due to the high cost and power consumption needed to process GPS signals.

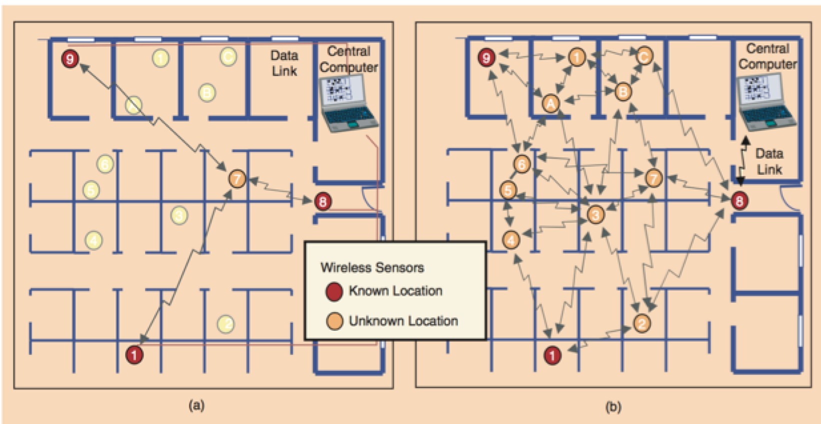

Figure 1.1: Difference between (a) traditional and (b) cooperative localization. The figure is taken from [1].

For localization to be possible, some nodes in the network, called anchor nodes, must know or have estimation of their positions. Those sensors need to obtain their coordinates from an external source (via GPS or from the network administrator). Then the rest of the sensors can often estimate their position by calculating the distance to the anchor nodes. To calculate the distance the devices must communicate between them. If sensors were capable of high-power transmission, they would be able to make measurements to multiple anchor nodes. However, energy-conserving devices like the one presented in [1] will not include a power amplifier and will be limited on battery and transmit power. Therefore, they would be only able to communicate with nearby sensors. To solve this, the localization techniques, will be multihop (a.k.a. collaborative localization). This means that unknown-location devices are still able to make measurements with anchor nodes. In addition, it is possible that unknown-location devices make measurements with other unknown-location devices. The additional information gained from these measurements enhances the accuracy and robustness of the localization system.

Figure 1.1 shows differences between traditional and cooperative localization. Figure 1.1(a) depicts a scenario where a sensor with unknown location (orange node indexed 7) estimates its location by using triangulation to three anchors (red nodes indexed 1, 8 and 9). Figure 1.1(b) depicts cooperative locations: measurements made between any pairs of sensors can be used to improve the location accuracy. For example sensor number 2 communicates with two anchor nodes (1 and 8) and with 2 unknown location nodes (3 and 7).

Figure 1.2 depicts how a localization system can be classified depending on its positioning technique, ranging technique or sensor device.

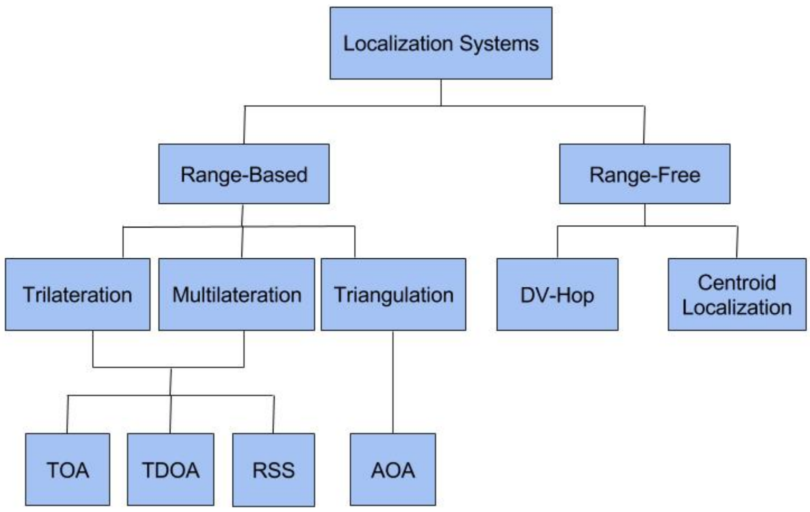

Figure 1.2: Classification of localization systems

1.1 Sensing

The selection of sensors is essential in the design of a WSN. Choosing the appropriate sensors for the application can improve the system's performance, lower its cost, and improve its lifetime. For localization, there is a wide range of sensors available [2], each one with different physical characteristics, performance and designed for different purposes. In the sequel, the most common sensors are reviewed.

1.1.1 Video Sensing

The first video cameras were completely analog [3]. This meant that the information was saved in cassettes (with a maximum length of 8 hours) without compressing. The next step was to digitalize the information and save it in hard disks. The information was compressed and the data from several cameras was multiplexed and stored together. Later it could be sent through Internet and visualized in another place. Then, in 1996, the first IP camera was released. It was a huge advance in video acquisition. There are many advantages of choosing IP instead of analog cameras: there is no more need of encoders because the video is digitalized inside the camera. Each camera is directly connected to Internet and has its own IP address. This provides more flexibility and remote accessibility, thus it can be seen live stream from any computer. They can work with PoE protocol; this means that they can work without a power supply, just with the alimentation provided by the Ethernet cable. Another advantage is that there is a two-way communication; this allows users to communicate with what they are seeing and also to transmit commands for PTZ (pan, tilt and zoom). Nevertheless there are some disadvantages: the cost is higher, require more bandwidth and as the information is transmitted over Internet it potentially becomes open to attacks by hackers.

1.1.2 Inertial Sensing

The Inertial Navigation Systems (INS) are devices that calculate the position, orientation and velocity of a moving object. Inertial Measurement Units (IMUs) typically contain three orthogonal rate-gyroscopes (measure the angular velocity) and three orthogonal accelerometers (measure the linear acceleration). To locate an object first it requires an initial position and velocity from another source (e.g. human or GPS satellite) and then itself computes and updates its position by integrating the information from the sensors.

1.1.3 Mechanical Sensing

Mechanical sensors are used to measure displacement, position, pressure, motion or flow. These are devices that change their behavior under the action of a physical force. Two types can be differentiated depending on their mode of operation. The first ones are based on the piezoresistive effect: their electric resistance is modified when a physical force is applied on them. They can be used to measure pressure. The second ones are based on the piezoelectric effect: convert a physical force to a difference in electrical potential.

1.1.4 Ultrasonic Sensing

Ultrasonic sensors generate high-frequency sound waves and then evaluate the echo. The distance to an object is determined measuring the time delay between the sent signal and the received echo. The acoustic signal can pass beyond small obstacles, but it is sensible to interferences and needs line-of-sight.

1.1.5 Optical Sensing

Optical sensors measure the amount of reflected or emitted light and translate it to an electronic signal. These systems have two components: light sources and optical sensors. Light sources can be passive objects that reflect the ambient light or active devices that emit internal generated light. The strong point is that they offer a long sensing range. However they need line of sight between the source and the sensor.

1.2 Ranging Techniques

Localization techniques are classified depending on how the measurements between a pair of sensors are obtained. This distance measurements can be attained from various signals, such as Received Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDoA) or Angle Of Arrival (AoA).

1.2.1 Received Signal Strength

Radio signal strength is defined as the power measured by a receiver. The RSS of acoustic, RF, or other signals can be considered. Wireless sensors communicate with neighboring sensors, so the RSS of RF signals can be measured by each receiver during normal data communication without presenting additional bandwidth or energy requirements. RSS measurements are relatively inexpensive and simple to implement in hardware. However, they are unpredictable due to the sources of error such as multipath and shadowing.

The value of the RSS is equivalent to the received power. The Friis equation gives the power received by one antenna in ideal conditions when another antenna transmits a known amount of power.

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi d} \right)^2 \tag{1.1}$$

Where P_r is the receiver's power (in watts); P_t is the transmitter's power; G_r and G_t the receiver and transmitter antenna gains; λ is the wavelength; d is the distance between transmitter and receiver.

The logarithmic expression of the free space loses is given by

$$FS(d) = 32.44 + 10 \cdot n \cdot \log(d) + 10 \cdot n \cdot \log(f) \tag{1.2}$$

Where FS(d) is the propagation path loss (in dB) after radio signal is transmitted d distance (expressed in Km); n is the signal attenuation coefficient (between 2 and 5 normally); f is the transmitted signal frequency (expressed in MHz).

Finally if equation 1.1 is expressed in dB, the P_r value satisfies

$$P_r = P_t + P_{\text{amplify}} - FS(d) \tag{1.3}$$

P_r is the received power, P_t is the transmitted power, $P_{amplify}$ is the gain of antennas (the sum of G_r and G_t), $FS(d)$ is the path loss. All the units in dB. Using equations 1.2 and 1.3, and knowing some values, the distance between two nodes can be easily calculated.

1.2.2 Time of Arrival

Time of arrival is the measured time at which a signal (RF, acoustic, or other) first arrives at a receiver. There are two types of ToA: one-way ToA and two-way ToA. One-way ToA in Figure 1.3(a), measures the signal propagation time between the transmitter and the receiver. They need to be synchronized with each other. Then the distance is calculated as

$$d_{i,j} = (t_2 - t_1) \cdot v_p \tag{1.4}$$

where t_1 and t_2 are the transmitter and receiver times; $d_{i,j}$ is the distance between them; and v_p is the signal's velocity of propagation. In this case the receiver calculates the distances that later uses to calculate the position.

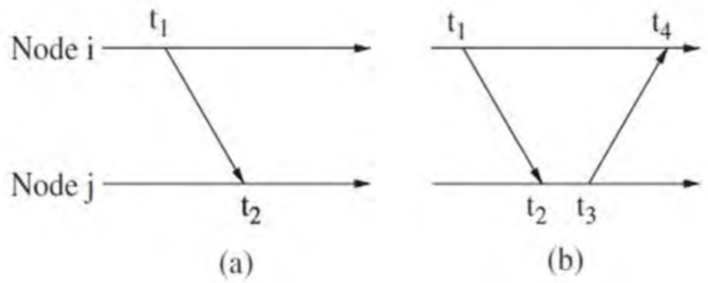

Figure 1.3: One-way and two-way ToA ranging measurement scheme. The figure is taken from [4].

In a two-way ToA in Figure 1.3(b), the receiver then sends a response signal back to the transmitter. Four times are used to calculate the distance:

$$d_{i,j} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \cdot v_p \tag{1.5}$$

where t_3 and t_4 are the transmitter and receiver times of the responding signal. As the transmitter is the one that finally calculates the distance, a third message is required to inform the receiver about the distance. In this case no synchronization is required.

The key element of time-based techniques is the receiver's ability to accurately estimate the arrival time of the line-of-sight (LOS) signal. This estimation is interfered both by additive noise and multipath signals.

1.2.3 Time Difference of Arrival

Time difference of arrival calculates the distance sending two signals at different speeds. As seen in Figure 1.4, at a time t_1 a signal with speed v_1 is sent. After a delay $t_{delay} = t_3 - t_1$ a second signal with a different speed v_2 is sent. Then the distance can be calculated like

$$d_{i,j} = (t_4 - t_2 - t_{delay}) \cdot (v_1 - v_2) \tag{1.6}$$

The benefit in front of TOA is that the transmitter and the receiver do not need to be synchronized.

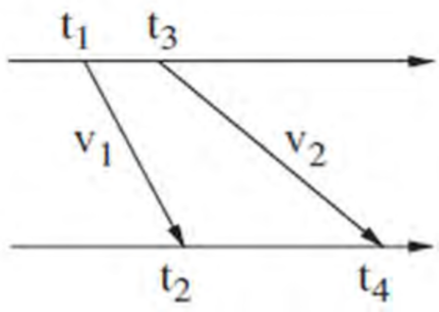

Figure 1.4: TDoA ranging measurement scheme. The figure is taken from [4].

1.2.4 Angle Of Arrival

Angle of arrival provides information about the direction to near sensors instead of the distance. The most common method is to use a sensor array and signal processing. In this case, each sensor node has two or more individual sensors (microphones for acoustic signals or antennas for RF signals) whose locations with respect to the node center are known. The AOA is estimated from the differences in arrival times for a transmitted signal at each of the sensor array elements. The estimation is similar to time-delay estimation discussed in the section on ToA but generalized to the case of more than two array elements. As explained in [5], one of the biggest drawbacks is that it requires multiple antenna elements, which increase sensor's cost and size. However, acoustic sensor arrays may already be required in devices and the use of MEMS and higher frequencies make the sensors smaller. The measurements are affected by additive noise and multipath.

1.3 Positioning

An unknown-location sensor, with a 2D localization, has to obtain 2 coordinates x and y:

$$\theta = \left[\theta_x, \theta_y\right] \tag{1.7}$$

In a 2D system, the process of estimation is less complex and requires less energy and time. It provides good accuracy on flat terrains but in hilly environments it is more difficult to estimate. By using 3-D localization one extra coordinate (θ_z) is required, the objective is to provide a more accurate result using height.

In a 2D space, three anchor nodes are required to determine the position. Knowing the distance between a terminal and an anchor node limits the target position to a circle, as shown in Figure 1.5(a). If the measurements of a second sensor are added, then the target position is reduced to two intersecting points Figure 1.5(b). A third sensor shows the final position in Figure 1.5(c).

In a 3D space, four anchor nodes are required. After the process explained before, the height of the target has to be determined with another measurement.

There are two types of localization: range-based and range-free. The first ones require distance measurements from other nodes to calculate its position. The second ones do not require distance measurement because they use connectivity information to determine its position.

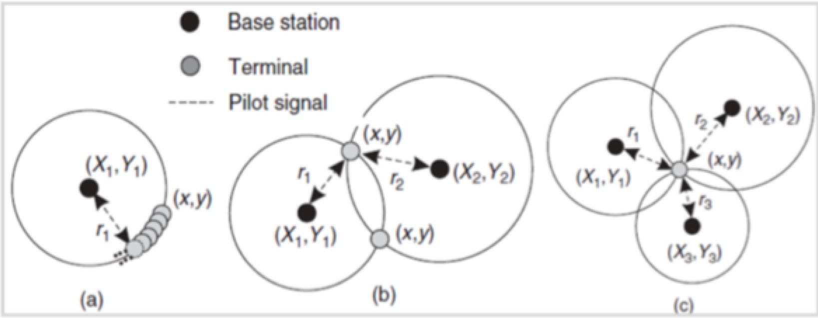

Figure 1.5: Example of localization. The figure is taken from [6].

1.3.1 Range-Based Localization

Triangulation

is the process of determining the location of a point by measuring angles of arrival from known points.

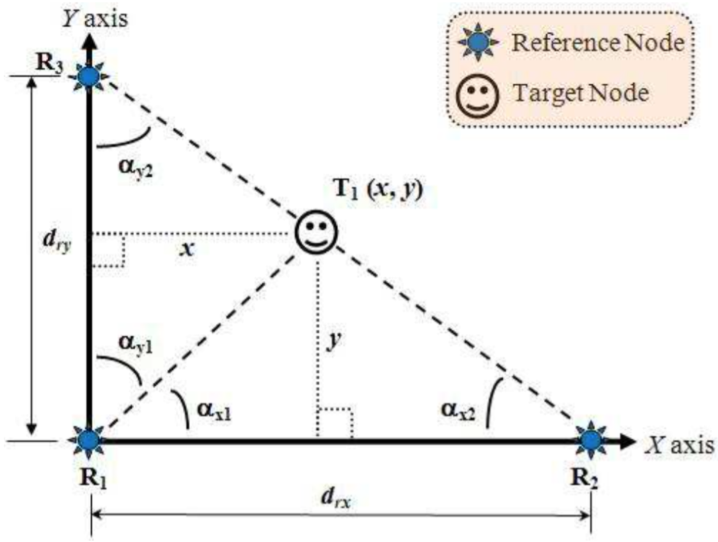

Figure 1.6: Example of triangulation. The figure is taken from [7].

In Figure 1.6 there are three anchor nodes (R_1, R_2 and R_3) and a target (T_1). Using the angles that they form [7], the coordinates (x, y) can be calculated:

$$x = \frac{d_{ry}\sin(\alpha_{y1})\sin(\alpha_{y2})}{\sin(\alpha_{y1} + \alpha_{y2})} \tag{1.8}$$

$$y = \frac{d_{rx}\sin(\alpha_{x1})\sin(\alpha_{x2})}{\sin(\alpha_{x1} + \alpha_{x2})} \tag{1.9}$$

Trilateration

is the process of determining a node's position based on the distances between this node and other nodes whose positions are known. In 2D geometry, it is known that if a point lies on two circles, then the circle centers and the two radii provide sufficient information to narrow the possible locations down to two. Additional information may narrow the possibilities down to one unique location. In 3D geometry, when it is known that a point lies on the surfaces of three spheres, then the centers of the spheres along with their radii provide sufficient information to narrow the possible locations down to no more than two.

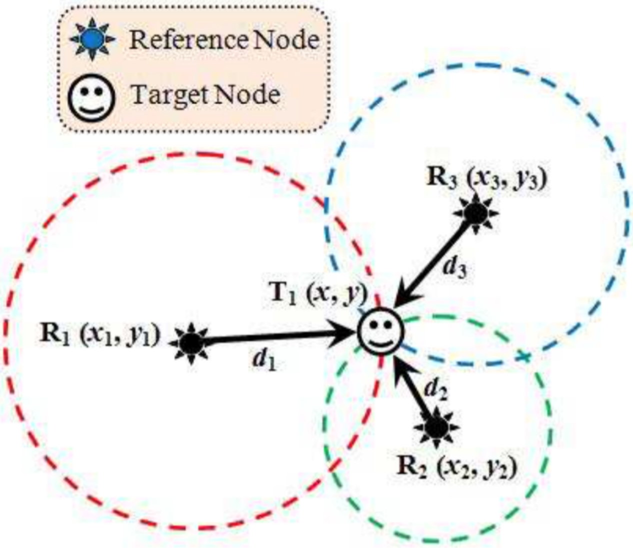

Figure 1.7: Trilateration estimation. The figure is taken from [7].

In Figure 1.7 there are three anchor nodes (R_1, R_2 and R_3) and a target (T_1). The distances (d_1, d_2 and d_3) between them can be calculated using Pythagorean theorem as in the following expressions from [7]:

$$\begin{aligned} d_1^2 &= (x_1 - x)^2 + (y_1 - y)^2 \\ d_2^2 &= (x_2 - x)^2 + (y_2 - y)^2 \\ d_3^2 &= (x_3 - x)^2 + (y_3 - y)^2 \end{aligned} \tag{1.10}$$

If the equations are reorganize and solved for x and y, the coordinates of the object can be obtained:

$$x = \frac{AY_{32} + BY_{13} + CY21}{2(x_1 Y_{32} + x_2 Y_{13} + x_3 Y_{21})} \tag{1.11}$$

$$y = \frac{AX_{32} + BX_{13} + CX21}{2(y_1 X_{32} + y_2 X_{13} + y_3 X_{21})} \tag{1.12}$$

where

$$\begin{aligned} A &= x_1^2 + y_1^2 - d_1^2 \\ B &= x_2^2 + y_2^2 - d_2^2 \\ C &= x_3^2 + y_3^2 - d_3^2 \end{aligned} \tag{1.13}$$

and

$$\begin{aligned} X_{32} &= (x_3 - x_2) \\ X_{13} &= (x_1 - x_3) \\ X_{21} &= (x_2 - x_1) \end{aligned} \tag{1.14}$$

$$
\begin{aligned}
Y_{32} &= (y_3 - y_2) \\
Y_{13} &= (y_1 - y_3) \\
Y_{21} &= (y_2 - y_1)
\end{aligned}
\tag{1.15}
$$

The position of an object can be determined just knowing the distances and the position of three anchor nodes.

Multilateration

is used in large scenarios where there are only some nodes that are equipped with GPS modules, so all the others require to locate only using those nodes. There are 3 possible scenarios:

1. The node can reach 3 GPS nodes.
2. The node can reach only one GPS node.
3. The node cannot reach any GPS node.

Only in the first case is possible to use lateration techniques, in the other two examples atomic and iterative multilateration [8] are used.

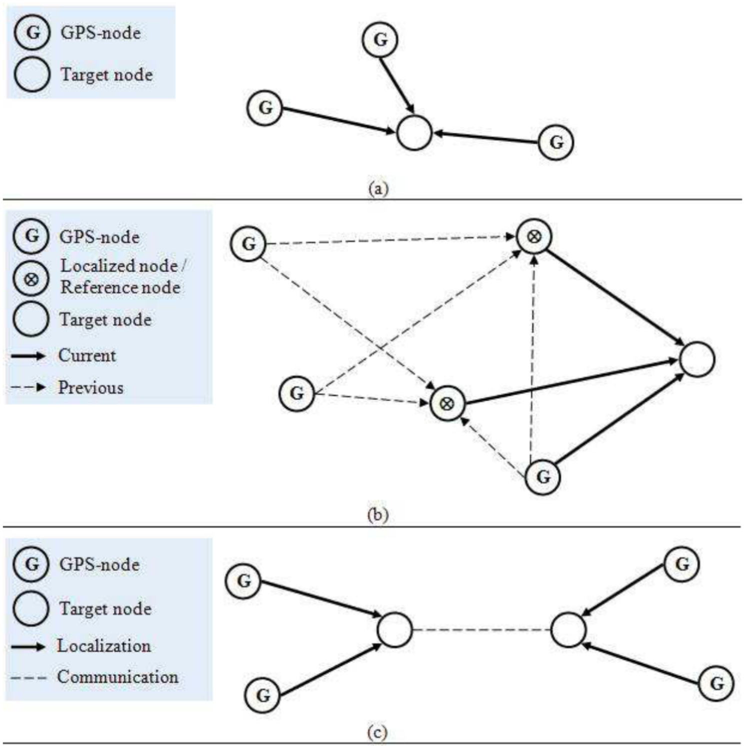

Figure 1.8: Atomic, Iterative and Collaborative Multilateration. The figure is taken from [7].

In atomic multilateration Figure 1.8(a) the location is estimated using three GPS-nodes. If the GPS nodes are too far and it is not possible to communicate with at least three of them, iterative localization is performed. In this case, sensor nodes are considered anchor nodes after being localized using GPS nodes as shown in Figure 1.8(b). Then, this new anchor nodes can be used to localize other nodes that are not possible to reach a GPS node. This action is continued until all nodes are localized. However, in large and spread WSN no sensor node can reach at least three GPS nodes at initial state. To solve this, collaborative localization in Figure 1.8(c) is proposed. Two unknown location nodes are close but are only able to reach two GPS nodes. They can communicate between them to obtain their positions.

1.3.2 Range-Free Localization

Centroid Localization

Centroid localization algorithm [9] broadcasts all possible node's location information to all other target nodes. Using that location information (x_i, y_i), the target nodes estimate their position (x_{target}, y_{target}):

$$(x_{target}, y_{target}) = (\frac{1}{N}\sum_{i=1}^{n} x_i, \frac{1}{N}\sum_{i=1}^{n} y_i) \tag{1.16}$$

where N is the total number of nodes used in the localization. However, this algorithm is not considered accurate because of the simplicity and approximations.

DV-Hop

DV-Hop considers hop counting to estimate the distance [7]. At the start all nodes broadcast their node ID and information to the nearest nodes and store a distance vector. Then each node diffuses this distance vector incrementing the hop count value. At the end all nodes have a distance vector of all the other nodes. The final step is to find the average distances using the following expression:

$$HopSize_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_j} \tag{1.17}$$

where $HopSize_i$ is the average single hop distance for a node i; (x_i, y_i) is the location of the node i; (x_j, y_j) is the location of the other nodes; h_j is the hop count distance from node i to node j. This algorithm works well if the distribution and density are good. However, if the deployment is not regular and there are not enough nodes the accuracy decreases.

In a real environment, signal propagation is affected by several sources of error, that consequently cause inaccuracies in the position estimation. In the following section will be discussed.

1.4 Sources of Error

In free space, signal power decays proportional to d^{-2}, being d the distance between the transmitter and receiver. This is the first cause of signal attenuation, but the two main sources of error are multipath signals and shadowing. Multipath occurs when two or more signal arrive to the receiver at a different times. The signals have different amplitudes and phases, and can be added constructively or destructively, causing frequency selective fading. Shadowing is the other main reason. Before reaching the receptor, a signal has to pass through many obstacles that cause attenuation. These attenuation is modeled as lognormal probability density function. Another source of error is the Additive White Gaussian Noise (AWGN). It is called white because affects the same way all the range of frequencies and Gaussian because it has a normal distribution in the time domain with average zero. It is modeled like this to simulate the effects of random sources that come from the nature.

Time-varying errors and environment-dependent errors can be distinguished. The first ones, include additive noise, multipath and interferences, can be reduced by averaging multiple measurements over time. The seconds, are the result of physical obstacles (walls, trees, people), and are unpredictable and modeled random. Usually those obstacles are stationary and constant over time.

Chapter 2

Wireless Camera Networks

A Wireless Camera Network (WCN) is a camera-based WSN that uses cameras as main sensors. It should have the same limited characteristics as a the sensor-based WSN: limited computational and data storage capacity, low communication bandwidth and low consumption. So, a centralized scheme in which all the information is sent and processed in a central node is not suitable because of the high bandwidth required. A good option is to choose a distributed scheme, where each camera has enough computational capabilities to capture, and process the images and then send only the relevant information to a central node. That node will be in charge to collect all the processed information and execute the data fusion algorithms. The main phases after the video acquisition will be explained: detection, tracking and classification.

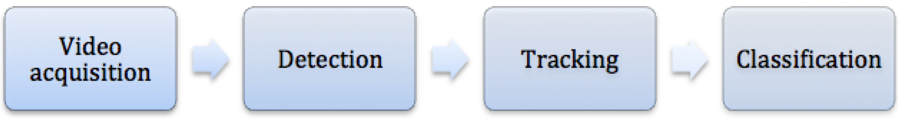

Figure 2.1: Phases of a wireless camera network

2.1 Detection

The first stage of a camera tracking system is to make a foreground estimation. In other words: to distinguish between the objects that are moving and the ones that remain static. As explained in [10], background subtraction techniques are a widely used approach. The main idea is to subtract the current frame from a background model. So the background model has to be a representation of the scene with no moving objects and must be regularly updated to adapt to varying luminance conditions and geometry changes. Background subtraction should segment objects of interest when they first appear (or reappear) in a scene and then adapt to sudden and gradual changes. It is also important to define an appropriate pixel level stationary criterion so when a pixel satisfies this criterion is declared background and ignored. There are different background subtraction techniques depending on the memory requirements, the computational complexity and the accuracy:

2.1.1 Frame Difference

It is simple and the main idea is to subtract two consecutive frames followed by a thresholding. The performance in general is poor but in some cases, if the threshold is correctly adjusted, can be good enough.

$$|frame(i) - frame(i-1)| \geq Th \tag{2.1}$$

2.1.2 Temporal Median Filter

The method presented in [11] uses the median value of the last N frames and w times the last computed median value. This combination increases the stability of the background model. However, the computational cost is high and requires a big buffer that depends on the number of images stored.

2.1.3 Running Gaussian average

Every pixel (i, k) of the image is modeled as a Gausian p.d.f.. The model implies that parameters (μ, θ^2) are to be computed based in past samples of the pixels. Instead of computing the p.d.f. from scratch every new time, a running average is computed as in 2.2 or the simplified version in 2.3 :

$$\mu(i, j, k) = \alpha I(i, j, k) + (1 - \alpha)\mu(i, j, k - 1) \tag{2.2}$$

$$\mu_t = \alpha I_t + (1 - \alpha)\mu_{t-1} \tag{2.3}$$

Where I_t is the pixel's current value; μ_t the previous average; α is the memory factor (between 0 an 1). The variance has also to be updated; the equation is similar to the one for updating the mean. It only requires the previous variance, the pixel mean value and the current pixel value:

$$(1 - \alpha)\alpha_{t-1}^2 + \alpha(I_t - \mu_t)^2 = \sigma_t^2 \tag{2.4}$$

Where I_t is the current frame; μ_t the image of mean pixel values; σ_t^2 the image of variance pixel values. Every time t a pixel I_t can be classified as a foreground pixel if the inequality in 2.5 holds. Otherwise it will be classified as background.

$$|I_t - \mu_t| > K\sigma_t \tag{2.5}$$

Comparing to the temporal median filter, the computational cost and the memory required are reduced.

2.1.4 Mixture of Gaussians (MoG)

Different background objects may appear at a same pixel (i, j): for example a pixel representing a building with tree leaves and branches in front. All models will adapt to this. However, the change may not be forever and appear faster than the algorithm update. In [12], a method based on the superposition of Gaussians is proposed. The model assumes that the probability of observing a certain pixel value, x, at time t may be modeled by means of a mixture of Gaussians:

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t} G(x_t - \mu_{i,t} \sum i, t) \tag{2.6}$$

Each K Gaussian only describes one of the foreground or background objects (K is usually between 3-5). To distinguish between foreground and background, all the distributions have to be ordered based on the ratio between their peak amplitude ($\omega_{i,t}$) and standard deviation (σ_i). According to 2.7, the first B distributions that satisfy the threshold T are classified as background.

$$\sum_{i=t}^{B} \omega_i > T \tag{2.7}$$

In the actual pixel try to find the Gaussian that approaches most to its value. Then update the mean and variance values of this Gaussian and increment its relative importance $w_{i,t}$. The memory required is K times the one in the running Gaussian average but the computational cost is the same. It is the most used method.

2.1.5 Kernel Density Estimation

The histogram of the time evolution of a pixel provides an estimation of the p.d.f. of that pixel. Given the p.d.f. of a background pixel in a position (x, y) of the image, it has to be decided if the current sample belongs or not to the background. The problem with the histogram approach is that if the number of pixels is limited the estimated p.d.f. is noisy (discrete quantification) and usually lacks of samples. In [13] an alternative idea is proposed: background p.d.f. is given as the superposition of Gaussians that guarantees a continuous version of the histogram.

$$P(x_t) = \frac{1}{n} \sum_{i=1}^{n} \eta(x_t - x_i \sum t) \tag{2.8}$$

A pixel is considered as foreground if:

$$Pr(X_t) < Th \tag{2.9}$$

The threshold (Th) used can be adjusted to obtain the desired result.

2.2 Tracking

When the detection is done, background and foreground are separated, so it is possible to distinguish the interest moving objects. Next step is to track the movement of those objects along the space. The main idea of video object tracking is to find the correspondence between detected objects in consecutive frames. But this is not an easy task because it will appear noise, clutter and occlusions between objects, the most critical issue.

2.2.1 Model-Based Tracking

As explained in [14] and [15] a good approach is to use model-based tracking. Besides the information obtained from the sensors, a pre-existing model of the object or description of the movement can be used to enhance the localization and tracking. In surveillance applications this is usually applied to cars and people. The first example in [16] is a traffic control application. It can detect and track vehicles using the beforehand knowledge of their shape and movement. The first step is to create a 3D model that can represent different types of vehicles. Then, create a motion model that describes the dynamic behavior of a vehicle without knowing about the intentions of the driver. Once the model is obtained, next step is to match it with the image data edge segments. The problem here is that the resolution of the camera has to be good and the edges and corners are short and difficult to detect. Also there are occlusions of objects, so this increases the complexity of the background estimation. In order to avoid incorrect matches because of the shadows of the vehicles, an illumination model is also included. This provides a geometrical description of the shadows of the vehicles projected onto the street plane. It makes possible to track vehicles in small areas.

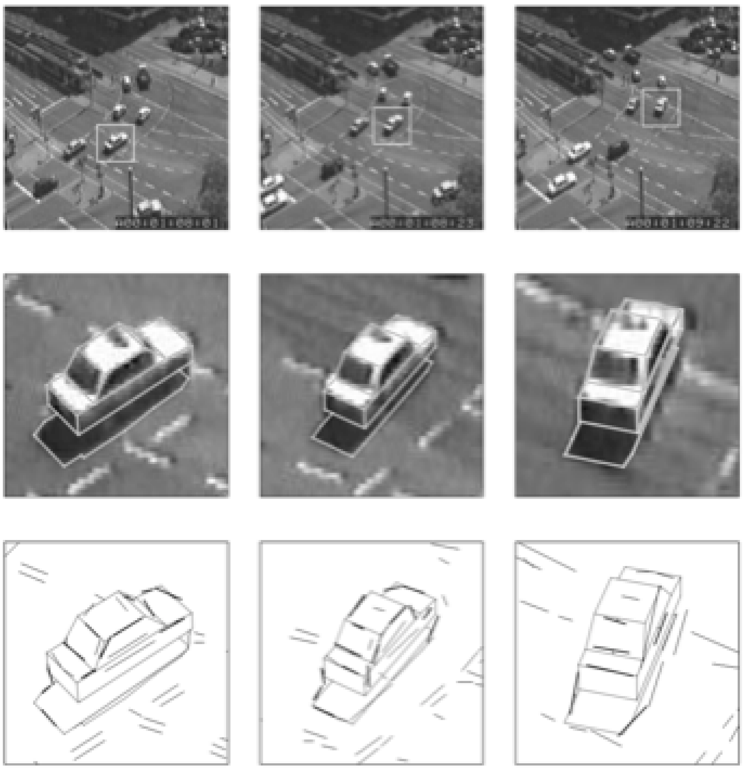

Figure 2.2: Example of results of model-based tracking. The figure is taken from [16].

The example in [17] tracks the motion and direction of a car in a road. Also, calculates the moving speed measuring the displacement of the target between two frames.

It is not only applied to cars. In [18] there is an example implemented in humans. The procedure is similar to the ones explained before: a body model is matched to the video images obtained. Then follows the tracking component with four main components included: prediction, synthesis, image analysis, and state estimation.

2.2.2 Tracking Filters

A common tracking method is to use a filtering mechanism to predict each movement of the recognised object. Kalman filter is the most frequently used filter. As explained in [19], the idea is that when there is a signal, there is also some noise. The simplest way to discard the noise would be to do an average of the samples, but in some cases it does not work well. So a more sophisticated technique should be used.

$$\hat{X}_k = K_k \cdot Z_k + (1 - K_k) \cdot \hat{X}_{k-1} \tag{2.10}$$

The purpose is to find $\hat{X}_k$, the estimated value of the signal x; K_k is the Kalman gain; Z_k is the measured value; $\hat{X}_{k-1}$ is the estimate of the signal on the previous state. The only unknown value is the Kalman gain (K_k), which is the key point.

At every k state a time update has to be applied (prediction)

$$\hat{x}_k^- = A\hat{X}_{k-1} + Bu_k \tag{2.11}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.12}$$

and also a measurement update (correction)

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{2.13}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.14}$$

$$P_k = (1 - K_k H)P_k^- \tag{2.15}$$

To start the process, it is necessary to know the estimate of x_0, and P_0.

2.2.3 Multiple Camera Tracking

In some cases [20] it is useful to use two or more cameras with overlapping views of the scene. This is for two reasons: the first is the use of depth information for tracking and occlusion resolution. The second is that using multiple cameras increases the area under view since it is not possible for a single camera to observe large areas because of a finite sensor field-of-view. It is necessary to compute the correspondence between objects of the different cameras. This process is accomplished by combining object appearance matching and camera geometry information. The detailed process is explained in [14].

2.3 Classification

The last step is to classify the objects analyzing its behavior. It consists in matching the measured sequence to a pre-compiled library that represent actions learned beforehand by the system using training sequences. According to [14] there are two approaches to object classification: image based and video based. The first one finds objects of a certain type without prior knowledge of the image location or scale. It is slower than video tracking based systems, which uses statistics about appearance, shape, and motion of items to distinguish what kind of object is.

Chapter 3

Fusion of Information

Visual detection and tracking provides really good results. However, there are some challenges that need to be solved: cameras are usually positioned in order to give a wide range of view, this provides low resolution images. In some cases the illumination is not appropriate, and may appear some shadowed areas. In real environments, there are also obstacles that cause target occlusions. Instead of trying to solve this using really sophisticated tracking algorithms a good solution is using fusion techniques. The objective is to improve the performance of the system and obtain more accuracy that could be achieved with the use of the sensors alone. Sensor fusion is the process of integrating raw and processed data from the different sensors in a central unit, the fusion center. Using combination of same source sensors, the performance is increased because of the redundant observation of a target. But it can be increased even more if different sensors devices are used.

The localization of an aircraft can be taken as a clear example [21]. A pulsed radar determines the aircraft's range, but has a limited ability to determine the angular. By contrast, the infrared imaging sensor can accurately determine the aircraft's angular direction, but is unable to measure range. When these two sensors are combined, the localization obtained is better than the one that could be achieved by either of the two independent sensors.

3.1 Architectures for Multisensor Data Fusion

A key issue in a multisensor data fusion system is to know at which point the data has to be combined. There are three architectures to fuse information:

3.1.1 Fusion of the Raw Observational Data

data from each sensor is aligned and transformed to convenient coordinates for central processing. Then the data is associated and correlated to determine which observations represent the same object. Once a determination has been made, then the data is fused, using sequential estimation techniques such as Kalman filters. This centralized fusion is theoretically the most accurate way to fuse data. However, this means that the raw data has to be transmitted from the sensors to the central processing. In the case of video images, it can suppose a high bandwidth.

3.1.2 Fusion of State Vectors

a state vector is an optimum estimate using an individual sensor's measurements of the position and velocity of an observed object. These vector estimates are then the input of the data fusion process. The association and correlation is still made but in a vector level. The communication between sensors and central node is reduced because the data now is compressed into a state vector. However, it is not as accurate as data level fusion because of the approximations made in the estimation of the state vectors.

3.1.3 Hybrid Approach

combines data level fusion and state vector fusion. State vector is performed to reduce the computational and communication demands. However, under required circumstances, data level fusion is performed.

3.2 Developed Systems

There are many examples in which video cameras are combined with different kinds of sensor devices. Just two of them will be explained.

In [22] a system called RAVEL (RAdio and Vision Enhanced Localization) is presented. It consists of two components: a visual based detector and a radio aided tracker. The camera captures a series of frames. The first assumption is that each frame contains a number of camera detections of moving objects represented as a bounding box of the detected object, or simply as the coordinates of the center of the bounding box. Also is assumed that at each time, the mobile device carried by a particular user receives a set of radio measurements of the RSS. The problem is to estimate the trajectory of a user given the sequence of anonymous camera detections and personal radio measurements. A Tracklet Generation Algorithm is used to find the correspondence of objects in consecutive frames. Also a Tracklet Merging Algorithm decides which tracklets should be merged together to produce the complete trajectory of the particular user.

The Swedish Defence Research Agency [23] proposes a system that provides accurate navigation in indoor environments without the need of a preinstalled infrastructure. It is used a combination of a foot-mounted inertial measurement unit (IMU) and a camera-based system with another IMU.

The Extended Kalman Filter (EKF) of both systems are fused into a new EKF where the state vector is

$$x = (p_f, v_f, \phi_f, p_c, v_c, \phi_c, a_c, g_c)^T \tag{3.1}$$

where p_f is the position of the foot-mounted system; v_f is its velocity; ϕ_f is the orientation; p_c is the position of the video system; v_c is its velocity; ϕ_c is the orientation; a_c and g_c are the accelerometer and gyro biases. Also three new equations that specify the relation between both systems are required. Results show that if there is sufficient daylight, the camera-based system provides good performance. However, in dark or smoke conditions or when steps can be distincted, the foot-mounted system provides the accuracy needed. Also when both work together the results are far better.

Chapter 4

Discussion and Novel Application

In many applications camera networks are only used for surveillance. In some cases this process is not even automatized, therefore, a person has to be watching a monitor in case something happens. The novel application proposed by the author improves those camera networks in museums, shops or supermarkets to analyze the movements patterns that follow the people.

It would not suppose a big cost, because usually in those places there is already a video surveillance system installed. Therefore, it would only be needed to implement the detection and tracking to localize and follow the people. Then, the core part of this application: the behavior classification to extract conclusions from the movements.

It could be used to increase the security in public places. In a museum for example, to know which are the most visited artworks and how many people visits each one. This kind of information would be useful to reorganize the museum and to improve the security in the most crowded places.

Similarly, it could be used in a supermarket or shop. Knowing the patterns that follow the buyers can be really useful for the marketing department in order to increase the sales.

Chapter 5

Simulations

5.1 Localization Accuracy

The purpose of this simulation is to use the ranging techniques and the localization algorithms explained in chapter 1, to analyse how the localization is performed. Then, quantify how much does the error influence in the location accuracy. As seen before in section 1.3, with only three anchor nodes it is possible to localize an object. The next steps are the basic simulation settings:

1. Define the coordinates of the anchor nodes and the track that is going to follow the target. The area is a square of 10.000 m^2.
2. Using the equations 1.2 and 1.3 and knowing the value of P_r, P_t, $P_{amplify}$ and f the distance to each anchor node can be easily obtained.
3. Using the equations 1.11, 1.12 and knowing the distances and position of the anchor nodes the final coordinates of the target are obtained.

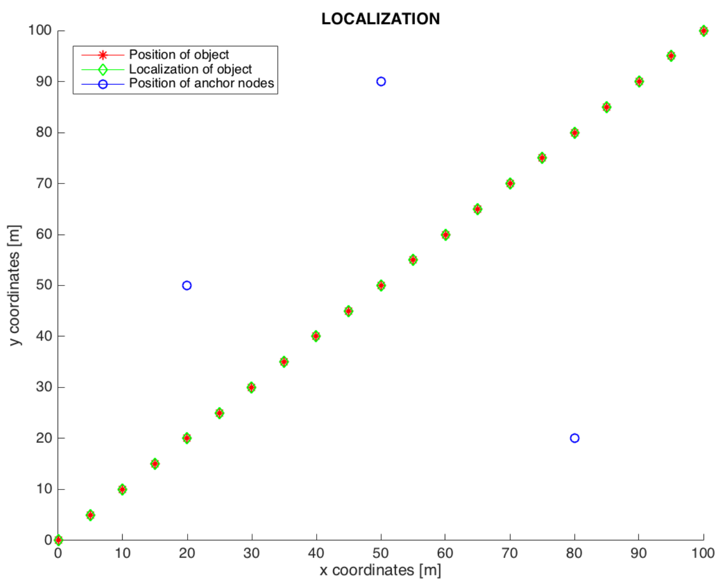

Figure 5.1: Localization without error using 3 anchor nodes.

Table 5.1: Localization error

Number of anchor nodes	3	5	10	25	50	100	500	1000
Mean error (m)	13,07	9,86	6,49	4,28	3,08	2,28	1,34	1,15
Reduction in %	-	-32,54	-52,06	-51,64	-38,85	-35,25	-70,13	-16,54
Deviation error (m)	4,60	2,20	1,11	0,53	0,42	0,23	0,11	0,05

Figure 5.1 depicts the case when there is no error. The location obtained is exactly the same as the position of the target. However, in a real environment there are sources of error that must be taken into account. When these loses are modeled the calculated distance will differ to the real one. AWGN with mean 0 will be generated and added in equation 1.3 like this:

$$P_r = P_t + P_{\text{amplify}} - FS(d) - AWGN \tag{5.1}$$

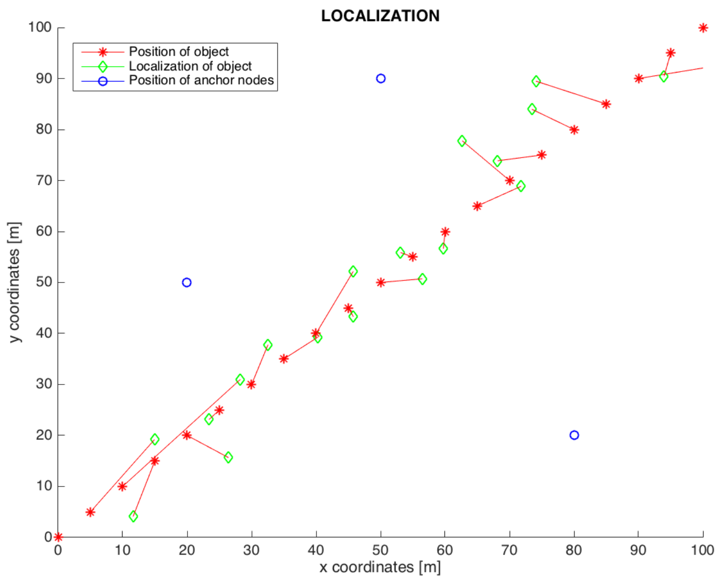

Figure 5.2: Localization with error using 3 anchor nodes.

In Figure 5.2, the inclusion of error generates inexactitudes that should be corrected. However, it is a simulation and it is not possible to measure the RSS, so a small change must be done. The procedure should be measure the RSS and calculate the distance. But the distance will be calculated (because the position of the anchor nodes and the target are defined) then $FS(d)$ and RSS. Now that the missing value is obtained, the AWGN can also be added and the distance calculated again.

What comes next is to a make it work with more than 3 anchor nodes. Equations 1.11 and 1.12, are only useful when there are 3 anchor nodes. The proposed change is to use the function $fminunc$ [24]. It finds the minimum of a problem specified by

$$\min_x f(x) \tag{5.2}$$

where $f(x)$ is a function that returns a scalar and x is a vector or a matrix.

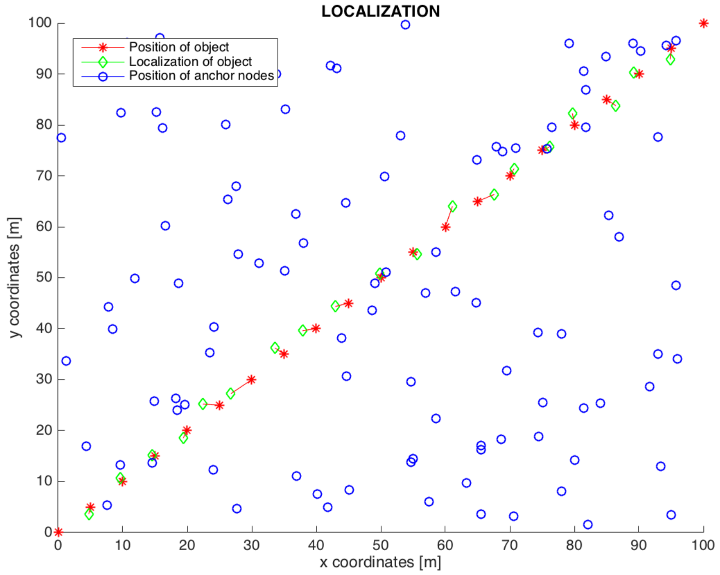

Figure 5.3: Localization with error and 100 anchor nodes.

In Figure 5.3 the same simulation is performed but using 100 anchor nodes instead. Comparing Figure 5.2 and Figure 5.3 it can observed that when the number of anchor nodes is increased, the error in distance localization is reduced. Related to this, another important analysis performed is how much does the number of nodes influence in the accuracy of the localization. To do this the error in each position of the target is calculated and then all values are averaged. It starts with the minimum number of nodes, 3, and then is increased it until 1000. In Table 5.1 can be seen the values of the mean error and deviation error, and also the error reduction in % for every increment depending on the number of anchor nodes. Figure 5.4, is

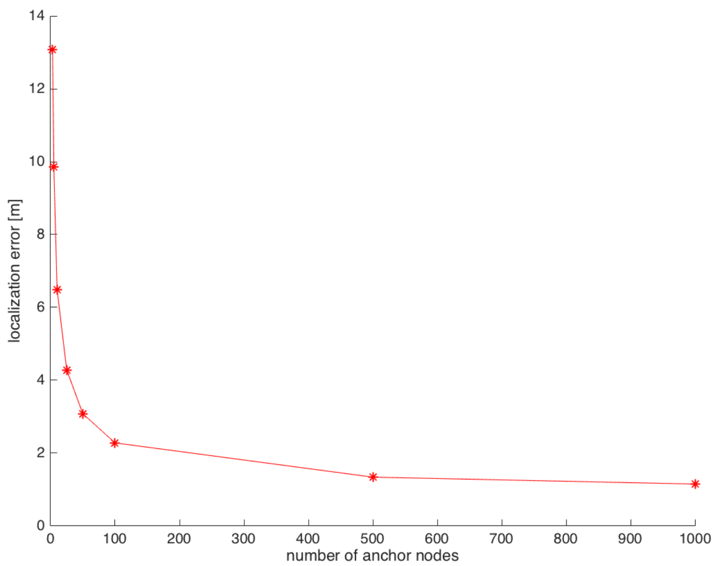

Figure 5.4: Localization error depending on the number of anchor nodes

the representation of the values from Table 5.1. The error is reduced a lot when increasing from 3 to 100 nodes. Then, every time a larger number of nodes is required to obtain more accuracy.

5.2 Multiple Object Detection and Tracking

The next Matlab simulation can be found in [25], shows how the detection and tracking of multiple moving objects is performed. To detect the objects a background subtraction algorithm based on the Mixture of Gaussians is used, then some filters are applied to reduce the noise, and finally a blob analysis detects groups of pixels that correspond to the same object. The results can be seen in Figure 5.5, where only the detected objects are displayed in white and with a box around and a number.

Figure 5.5: Background subtraction of objects using MoG.

After the detection, the Kalmar filter is used to predict the track's location in each frame and update the bounding box. It is also important to keep updated the track assignment. The assigned tracks are updated using the detections and the unassigned tracks are marked invisible.

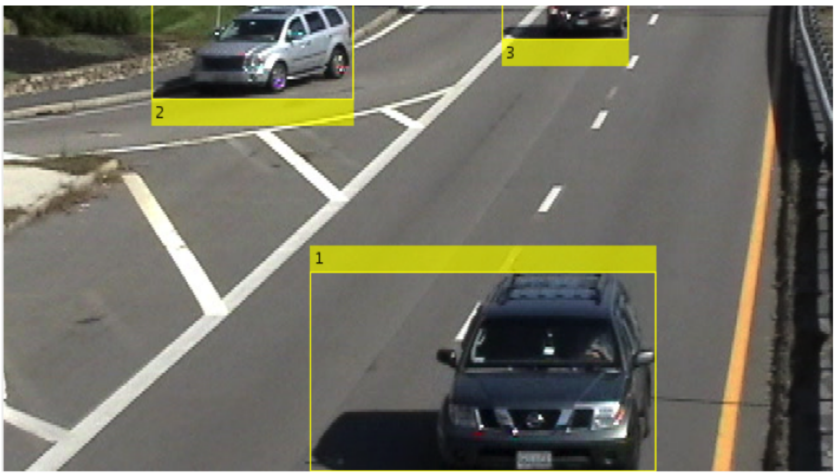

Figure 5.6: Multiple cars detection and tracking.

In Figure 5.7 another simulation is performed but instead of using traffic data, a video of people walking is used.

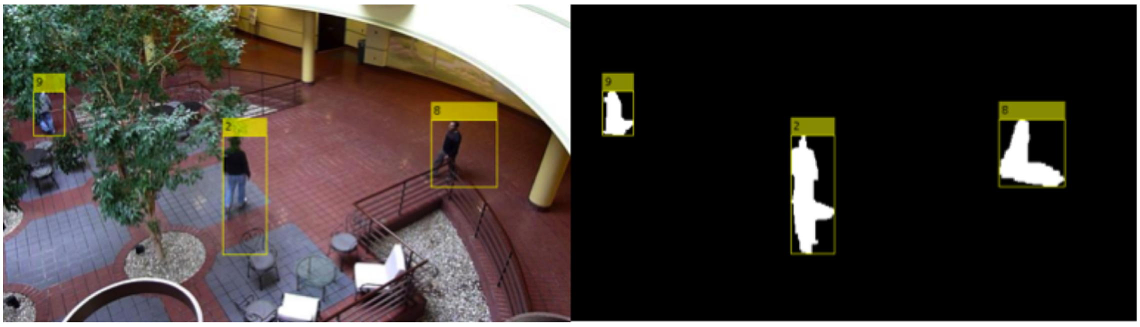

Figure 5.7: Multiple people detection and tracking.

Chapter 6

Conclusions

This paper has presented some of the essential characteristics of the wireless sensor networks, particularly the camera based WSN. Cameras are among the best kind of sensors to detect and track objects because of the high amount of information they generate. However, other sensors should be used at the same time to improve its performance. Since there is a broad range of sensors, the most appropriate needs to be chosen depending on the application. In object detection there are different methods for background subtraction. The MoG method is the one that gives the best results, even though it demands more computational requirements. Then for the object tracking, the most used method is the Kalman Filter. It is also interesting to use model-based tracking to increase the exactitude when having prior knowledge of the target. Examples of fusion techniques are also shown. The gain is not only in accuracy, but also in reliability of the system. Finally two simulations using Matlab are made. In the first one, the localization of a target is performed. When increasing the number of anchor nodes, the error in target localization is reduced. The second one is a revision of the detection and tracking in order to have a visual and clearer idea of the results obtained.

Chapter 7

Future Work

Although wireless camera networks and WSN have been well studied, still improvements can be done. For example, when an object is moving and goes through different cameras there has to be a way to find the correspondence between the cameras. Therefore, a handover tracking algorithm across cameras should be defined. A key issue to build automated surveillance systems is the interpretation of the behaviour of the recognised objects. Even if the best detection and tracking are performed, if then the extraction of information is not good the system will not be autonomous. Also it is interesting that the system itself calls an emergency number if a specific alarm has been detected. Dealing with PTZ cameras in order to view wider areas and obtain better quality images. The creation of metadata standards to solve the bandwidth limitations. In the same way to work on new protocols for distributed architectures and real-time communications.

Bibliography

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal. "Locating the nodes: cooperative localization in wireless sensor networks". In: *IEEE Signal Processing Magazine* 22.4 (2005), pp. 54–69. ISSN: 1053-5888. DOI: 10.1109/MSP.2005.1458287.

[2] G. Welch and E. Foxlin. "Motion tracking survey". In: *IEEE Computer graphics and Applications* (2002), pp. 24–38.

[3] Branimir Jaksic Boris Gara Gradimirka Popovic Nebojsa Arsic and Mile Petrovic. "Overview, Characteristics and Advantages of IP Camera Video Surveillance Systems Compared to Systems with other Kinds of Camera". In: *International Journal of Engineering Science and Innovative Technology* 2.5 (2013), pp. 356–362.

[4] Carlo Fischione. *An Introduction to Wireless Sensor Networks*. Sweden: KTH, 2015.

[5] Minghui Li and Yilong Lu. "Angle-of-arrival estimation for localization and communication in wireless networks". In: *Signal Processing Conference, 2008 16th European*. 2008, pp. 1–5.

[6] Mohamed H Abdel Meniem, Ahmed M Hamad, and Eman Shaaban. "Fast and Accurate Practical Positioning Method using Enhanced-Lateration Technique and Adaptive Propagation Model in GSM Mode". In: *International Journal of Computer Science Issues* 9.2 (2012), pp. 188–193.

[7] Pu C. H. Pu C. C. and H. J. Lee. *Indoor location tracking using received signal strength indicator*. Rijeka, Croatia: INTECH Open Access Publisher, 2011.

[8] Andreas Savvides, Chih-Chieh Han, and Mani B Strivastava. "Dynamic fine-grained localization in ad-hoc networks of sensors". In: *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM. 2001, pp. 166–179.

[9] N. Bulusu, J. Heidemann, and D. Estrin. "GPS-less low-cost outdoor localization for very small devices". In: *IEEE Personal Communications* 7.5 (2000), pp. 28–34. ISSN: 1070-9916. DOI: 10.1109/98.878533.

[10] M. Piccardi. "Background subtraction techniques: a review". In: *Systems, Man and Cybernetics, 2004 IEEE International Conference on*. Vol. 4. 2004, 3099–3104 vol.4. DOI: 10.1109/ICSMC.2004.1400815.

[11] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. "Detecting moving objects, ghosts, and shadows in video streams". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.10 (2003), pp. 1337–1342. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2003.1233909.

[12] C. Stauffer and W. E. L. Grimson. “Adaptive background mixture models for real-time tracking”. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* Vol. 2. 1999, 252 Vol. 2. DOI: 10.1109/CVPR.1999.784637.

[13] D. Harwood A. Elgammal and L. Davis. “Non-parametric model for background subtraction”. In: *Computer Vision—ECCV 2000.* Springer, 2000, pp. 751–767.

[14] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. “Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking”. In: *IEEE Signal Processing Magazine* 22.2 (2005), pp. 38–51. ISSN: 1053-5888. DOI: 10.1109/MSP.2005.1406476.

[15] M. Valera and S. A. Velastin. “Intelligent distributed surveillance systems: a review”. In: *IEE Proceedings - Vision, Image and Signal Processing* 152.2 (2005), pp. 192–204. ISSN: 1350-245X. DOI: 10.1049/ip-vis:20041147.

[16] D. Koller, K. Daniilidis, and H. H. Nagel. “Model-based object tracking in monocular image sequences of road traffic scenes”. In: *International Journal of Computer 11263on* 10.3 (1993), pp. 257–281.

[17] Z. H. Xi and G. H. Dong. “Moving Objects Detection and Speed Estimation in Intelligent Surveillance System”. In: *Advanced Materials Research.* Vol. 712. Trans Tech Publ. 2013, pp. 2354–2358.

[18] D. M. Gavrila and L. S. Davis. “3-D model-based tracking of humans in action: a multi-view approach”. In: *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR ’96, 1996 IEEE Computer Society Conference on.* 1996, pp. 73–80. DOI: 10.1109/CVPR.1996.517056.

[19] Bilgin Esme. *Kalman Filter For Dummies.* Available: http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies. March, 2009.

[20] O. Javed A. Yilmaz and M. Shah. “Object tracking: A survey”. In: *Acm computing surveys (CSUR)* 38.4 (2006), p. 13.

[21] David L Hall and James Llinas. “An introduction to multisensor data fusion”. In: *Proceedings of the IEEE* 85.1 (1997), pp. 6–23.

[22] S. Papaioannou, H. Wen, A. Markham, and N. Trigoni. “Fusion of Radio and Camera Sensor Data for Accurate Indoor Positioning”. In: *Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on.* 2014, pp. 109–117. DOI: 10.1109/MASS.2014.52.

[23] Erika Emilsson and Joakim Rydell. “Sensor fusion for improved indoor navigation”. In: *SPIE Security+ Defence.* International Society for Optics and Photonics. 2012, pp. 85420M–85420M.

[24] MathWorks. *fminunc.* Available: http://se.mathworks.com/help/optim/ug/fminunc.html. 3-29-2016.

[25] MathWorks. *Motion-Based Multiple Object Tracking.* Available: http://se.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html. 4-15-2016.